Drones and Autonomous Vehicles

SCIEPublish

Communication

A Lightweight and Efficient Authentication Scheme for the Internet of Drone Based on Cancelable Biometrics

Kanyi Chen [1,2], Weixin Bian [1,2,*], Qingde Li [3], Dong Xie [1,2] and Jinbin Meng [1,2]

[1] School of Computer and Information, Anhui Normal University, Wuhu 241002, China; chenkangyi@ahnu.edu.cn (K.C.); xiedong@ahnu.edu.cn (D.X.); 2421012576@ahnu.edu.cn (J.M.)

[2] Anhui Province Key Laboratory of Industrial Intelligence Data Security, Wuhu 241002, China

[3] Computer Science, School of Digital and Physical Sciences, University of Hull, Hull HU6 7RX, UK; q.li@hull.ac.uk (Q.L.)

* Corresponding author. E-mail: bwx2353@ahnu.edu.cn (W.B.)

Received: 10 January 2026; Revised: 30 January 2026; Accepted: 12 March 2026; Available online: 1 April 2026

ABSTRACT: Unmanned aerial vehicles (UAVs is also known as drones) have significant applications in smart cities, and the information exchange between UAVs and the control server (CS) is conducted through wireless communication channels, which are susceptible to various security risks, such as network attacks and drone capture. To ensure the security and integrity of information in the Internet of Drones (IoD), identity authentication and key agreement protocols can be designed for protection. However, due to the unique characteristics of IoD, such as the extremely high mobility of drones in real scenarios and the resource constraints of drones, there is a need to meet the requirements for lightweight protocols. This paper proposes a strategy that uses cancelable biometric features to protect the biometric features of users during the authentication process. The method combines Fast Fourier Transform, Gaussian random projections, Position-Sensitive Hashing, fuzzy extractors, and Physical Unclonable Functions (PUF), meeting the security and lightweight needs of IoD authentication protocols. We use the Real-or-Random (ROR) model and the Avispa simulation tool to prove that our protocol is secure. Through comparative research, the proposed cancelable method has higher matching efficiency and better unlinkability, and our protocol offers higher security and faster computational efficiency.

Keywords: Internet of Drones (IoD); Cancelable biometrics; Physical unclonable functions; Efficient and security

1. Introduction

The Internet of Drones (IoD) is a layered architecture that enables coordinated access and communication for Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, within a regulated airspace. A typical IoD system comprises drones, ground stations (GSs), and remote users, forming an interconnected ecosystem for aerial operations [1]. Initially deployed in military contexts, drones have since expanded into a broad range of civilian and industrial applications due to the miniaturization of sensors, advances in wireless communication, and growth in autonomous control technologies.

IoD applications now span urban traffic control, environmental monitoring, search and rescue operations, agriculture, industrial inspection, logistics, public safety, and even emergency healthcare delivery [2]. By equipping drones with high-precision sensors and integrating big data analytics and artificial intelligence (AI) capabilities, UAVs can perform complex tasks such as autonomous navigation, real-time anomaly detection, and cooperative swarm coordination with minimal human intervention. This improves operational precision, reduces human error, and enhances responsiveness in time-sensitive scenarios.

From a network perspective, the Air Traffic Networks (ATNs) offer significantly greater spatial freedom than traditional Ground Transportation Networks (GTNs). The rational exploitation of ATNs can relieve congestion in GTNs and open new paradigms for real-time, on-demand services in smart cities. According to Drone Industry Insights, the global drone market is projected to reach $54.6 billion by 2030, with a compound annual growth rate (CAGR) of 7.7%, indicating a robust upward trajectory.

However, the rapid proliferation of IoD technologies introduces critical challenges in security and privacy protection [3]. Due to their reliance on open wireless communication, drones are vulnerable to eavesdropping, jamming, replay attacks, spoofing, and signal injection. GPS spoofing can mislead navigation, and vulnerabilities in Wi-Fi protocol stacks create opportunities for unauthorized access. Moreover, drones often communicate with ground stations and remote servers over insecure channels such as Wi-Fi or Long Term Evolution (LTE), where conventional security assumptions may no longer hold.

To mitigate these threats, robust authentication, key agreement, and data integrity mechanisms must be integrated into the IoD framework. In response to these challenges, this paper proposes a multi-factor authentication and key agreement scheme, which incorporates Cancelable Biometric techniques and a Physically Unclonable Function (PUF) update mechanism. The approach aims to strengthen identity verification, support revocability of compromised biometric templates, and enhance resilience against physical and cyber-attacks. The scheme is designed to provide mutual authentication, session key establishment, and protection against impersonation, replay, and template inversion attacks, thereby achieving a balance between lightweight performance and high security assurance in IoD environments.

1.1. Related Work

A key component of securing IoD systems is robust authentication. However, many authentication and key agreement (AKA) protocols neglect biometric protection. Remote biometric authentication systems often face four main threats: sensor spoofing, database attacks, comparator attacks, and channel interception [4]. Cancelable biometrics enhance system security by transforming original biometric templates into protected representations in a secure domain, preventing the leakage of original samples [5]. Comparisons are performed within this domain to ensure data protection during identification.

Traditional authentication relies on passwords, which burden users and are prone to guessing attacks [6]. Therefore, we adopt biometrics, which serve as unique, hard-to-replicate identifiers [7]. Teoh [8] proposed Multispace Random Projections (MRP), a widely used method for cancelable template generation [9]. However, stolen projection matrices can allow reconstruction of the original biometrics. Index-of-Max Hashing (IoM) [10] is a classical method that maps biometric features into index-domain hash codes using gaussian random projection (GRP) or uniformly random permutation (URP) to generate secure templates. Kuzu et al. [11] applied deep learning to create cancelable finger vein templates, showing high robustness. Cancelable biometrics exhibit cancelability (template regeneration via parameter change), non-invertibility (blocking reconstruction), diversity (preventing cross-system linkability), and accuracy (maintaining recognition performance) [7].

Authentication schemes have evolved to multi-factor approaches that combine biometrics, smart cards, and auxiliary data for enhanced security [12]. A secure AKA protocol is crucial for pre-communication key agreement over insecure channels, especially in UAV networks. In 2021, Srinivas et al. [13] proposed UAP-BCIoT, a three-factor authentication protocol enabling mutual authentication and key agreement for smart

transportation. Sarier [14] proposed MFBA, integrating zero-knowledge proofs and homomorphic encryption to perform biometric comparison in encrypted space. However, storing transformation parameters on the smart card risks biometric leakage.

Recently, drone communications over wireless channels have gained attention. However, these channels expose drones to security threats [15]. Unlike traditional authentication protocols, our design supports dynamic key generation during mutual authentication. Gope et al. [16] introduced a PUF-based authentication with an update mechanism, enhancing resistance against attacks. Their scheme leverages physical-layer security for sensor networks. Due to limited drone resources, many lightweight protocols [17] sacrifice robustness. Liu et al. [18] proposed a PUF-based lightweight protocol without challenge-response pairs (CRP) update, while others adopt ECC and hash functions for efficient key management [19]. Despite PUF's popularity in IoT and IoD, few protocols integrate it with biometric protection. Given PUF's support for irreversibility, unlinkability, and reproducibility (key cancelable biometric requirements), this combination is promising. Bian et al. [20] confirmed its feasibility by integrating PUF and fuzzy extractors for secure template generation.

Table 1 presents the Gap Analysis and provides brief descriptions of 11 identity authentication-related research papers. The columns include Scheme/Year, Template Protection, Cancelable Property, PUF Usage, CRP Updateability, Formal Proof (ROR Model), Formal Verification (AVISPA Tool), Attack Coverage, and Dataset-Based Evaluation. This structure renders the novelty, security, and efficiency claims of the proposed scheme traceable and persuasive.

Table 1. Gap Analysis.

Scheme/Year	Template Protection	Cancelable Property	PUF Usage	CRP Updateability	ROR	AVISPA	Attack Coverage	Dataset-Based Evaluation
Das [12]/2009	Unused	No	Unused	No	Unused	Unused	A4–12	Unused
Srinivas [13]/2021	Primitive Biometric	No	Unused	No	Yes	Unused	A1–12	Unused
Sarier [14]/2010	Cancelable Biometric	No	Unused	No	Unused	Unused	A4–8, A10, A11, A13	Unused
Kirsal Ever [15]/2019	Unused	No	Unused	No	Unused	Unused	A1, A2, A4–12	Unused
Gope [16]/2019	Primitive Biometric	No	Yes	No	Yes	Unused	A1–12	Unused
Chaudhary [17]/2023	Unused	No	Unused	No	Yes	Unused	A1–12	Unused
Liu [18]/2020	Primitive Biometric	No	Yes	No	Yes	Unused	A1, A3–12	Unused
Tanveer [19]/2022	Primitive Biometric	No	Unused	No	Yes	Yes	A1–12	Unused
Bian [20]/2020	Primitive Biometric	No	Yes	No	Unused	Unused	A1, A2, A4–8, A10, A11	Unused
Zhang [21]/2021	Cancelable Biometric	Yes	Yes	No	Yes	Unused	A1–13	FVC
Hu [22]/2024	Cancelable Biometric	Yes	Yes	Yes	Yes	Unused	A1–13	FVC

A1: User Anonymity; A2: Forward Secrecy; A3: Sensor Anonymity; A4: Replay Attack; A5: Man-in-the-Middle Attack; A6: Physical Attack; A7: Device Loss Attack; A8: Known session Key attack; A9: Password Guessing Attack; A10: User Device Loss Attack; A11: Offline Attack; A12: Sensor Theft Attack; A13: Biometric Attack; FVC: Fingerprint Verification Competition Datasets.

1.2. Motivation and Contribution

Addressing the various limitations of identity authentication in IoD, we have designed a new protocol to meet the challenges. In this protocol, we employ a cancelable biometric template protection method to safeguard biometric data. Specifically, the contributions of our paper are as follows:

(1) Combined with Fast Fourier Transform (FFT), IoM hash, Fuzzy extractor (FE), and Physical Unclonable Function (PUF), this paper proposes a better method for generating cancelable biometric templates.
(2) This paper proposes a lightweight mutual authentication protocol using cancelable biometrics with PUF updates and a single challenge-response pair (CRP), optimized for resource-limited IoD.
(3) Provide a formal security analysis based on the ROR model. Additionally, the security analysis indicates that the protocol can resist many attacks.

The rest of this article is organized as follows: Section 3 provides preliminary information. Section 4 describes the proposed template protection process and the proposed authentication scheme. Section 5 analyzes the security performance of the proposed scheme. Section 6 describes the experiments performed and the associated performance analysis. This article is summarized in Section 7.

2. System Model

2.1. Network Model

This section describes the network model and the adversary model in detail.

As shown in Figure 1, the network architecture is composed of three parts: (i) users (U_i), equipped with mobile devices integrated with biometric scanners (e.g., fingerprint sensors) and smart cards (SC) for local storage of authentication parameters $(A, ID'_{cs}, RPW, \tau_i)$. (ii) Control center server (CS), a fully trusted entity responsible for generating global keys K_{global}, challenges (C_u, C_d), pseudo-identities (PID_u, PID_d), and storing registration data $(PID_u, (C_u, R_u))$, $(ID_d, PID_d, (C_d, R_d))$. and (iii) Drones ($RD_j$), resource-constrained devices that register with CS to obtain authentication credentials, store PID_d locally, and execute PUF-based challenge-response mechanisms. The operator must carry a mobile device to receive data and store it locally for use during registration or mutual authentication processes. The CS is a trusted entity that generates auxiliary parameters for both U_i and RD_j; it is also needed to store the registered IDs or pseudo-IDs of U_i and RD_j.

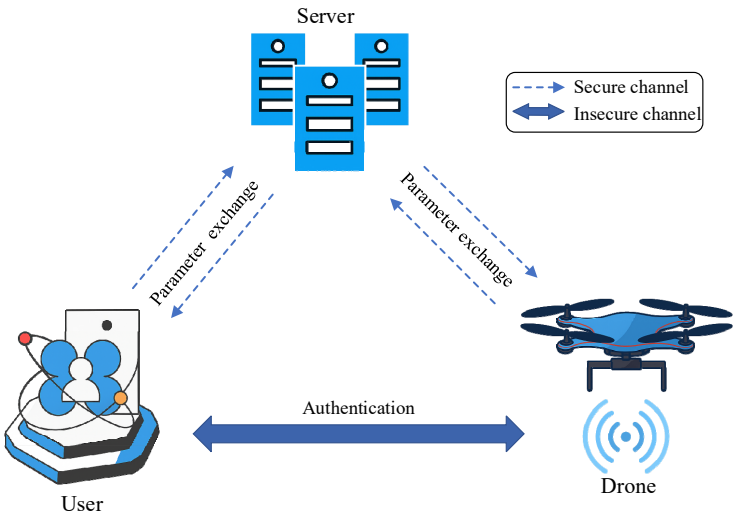

Figure 1. Network model of IoD.

The drone also needs to register with the *CS* to obtain key data for mutual authentication. If U_i and RD_j successfully complete the session key agreement through the *CS*, and a temporary session key will be generated for secure data communication.

2.2. Adversary Model

The core assumptions of the system are as follows: Communication between users (U_i), the control server (*CS*), and drones (D_j) is conducted via insecure public channels such as Wi-Fi and LTE, which are vulnerable to interception and tampering; user smart cards and drones are untrusted and may be lost or stolen, and the data stored locally on them can be extracted through side-channel attacks (e.g., power analysis); the control server is immune to all attacks and maintains the data integrity of the stored registration information.

This article adopts the widely used Dolev–Yao (DY) threat model for analyzing the security of cryptographic protocols [23]. Our protocol involves three types of participants: users (U_i), the control server (*CS*), and drones (RD_j), each with multiple instances. The DY model assumes that an adversary $\mathcal{A}$ having multiple instances:

(1) Public Channel Control: Intercept, modify, delete, inject, or replay messages transmitted over public channels (supporting replay attacks and man-in-the-middle attacks), while being unable to access information exchanged via secure channels.
(2) Compromised Entity Exploitation: The *CS* is fully trusted and immune to attacks, but other entities (users and drones) may be compromised; user smart cards (SC) may be lost or stolen, with sensitive data $(A, ID'_{cs}, RPW, \tau_i)$ extractable via side-channel attacks (e.g., power analysis), and the drone's memory can be compromised to obtain stored PID_d and historical Challenge-Response Pairs (C_d, R_d).
(3) Parameter Leakage: Partially acquire transformation parameters (e.g., partial Gaussian random projection matrices, random permutation arrays r) but not the complete set.
(4) Tampering Attacks: Manipulate timestamps (t_1–t_6) in authentication messages to bypass freshness checks; forge PUF responses if CRPs are leaked.
(5) Guessing Attacks: Launch online or offline password guessing attacks using a finite password dictionary (DC) and a biometric feature space.

3. Preliminaries

3.1. Fuzzy Extractor

Fuzzy extractor (FE) is a cryptographic tool used to extract and reproduce a reliable and random key from noisy data. There are two functions, *FE.Gen*(·) and *FE.Rep*(·) [20]. Enter the biometric *Bi* to *FE.Gen*(·) to generate auxiliary data σ and key τ, denoted as (σ, τ) = *FE.Gen*(*Bi*). If the subsequently extracted biometric is sufficiently similar to *Bi*, the σ can be recovered by *FE.Rep*(Bi', τ). FE is not sensitive to noise, as long as $\mathrm{dis}(Bi, Bi') < t$, the same (σ, τ) can be generated [21]. FE can effectively address the issue of minor differences in biometric features during the registration and authentication phases.

3.2. Cancelable Biometric

Cancelable biometrics enhance the security and privacy of biometric systems by transforming the original biometric vector $\mathbf{x} \in \mathbb{R}^n$ into a protected template $\mathbf{y}$ using a user-specific transformation function T_K, such that $\mathbf{y} = T_K(\mathbf{x})$, where K is a secret key. When a template is compromised, a new unlinkable template can be generated by altering the key, *i.e.*, $\mathbf{y}' = T_{K'}(\mathbf{x})$ with $K' \neq K$. This approach ensures cancelability, non-invertibility, diversity, and matching accuracy: the original biometric cannot be feasibly recovered from $\mathbf{y}$; different keys produce distinct templates for the same biometric input, *i.e.*, $T_{K_1}(\mathbf{x}) \neq$

$T_{K_2}(\mathbf{x})$ for $K_1 \neq K_2$; and the transformation preserves recognition performance within the transformed domain [4,7].

3.3. Physically Unclonable Function

Physical Unclonable Function (PUF) are a hardware security technology based on the physical characteristics of a device. PUF comes from the IC manufacturing process, which adds random physical changes to the IC's microstructure, making it unique and impossible to clone [24]. The core concept of PUF is CRP: a random string is input, and the PUF generates a unique response. PUFs are unpredictable but easy to build and check, making them suitable for security protection in lightweight devices. Since the output of a PUF is physically dependent, any tampering will alter its behavior, making it useless. These features enable PUF to be used as a cancelable step or as an authentication factor, increasing system complexity and improving security at the physical level [20].

4. Proposed Scheme

Table 2 lists the important symbols used to describe the proposed scheme. In the proposed scheme, the user's fingerprint is captured and extracted by the device for local authentication, and cancelable biometric methods are used to avoid the risk of permanent leakage of biometric information. Currently, most mobile devices can be equipped with biometric scanners, which can easily scan the user's fingerprint onto the device, thus solving the problem of biometric acquisition in the proposed scheme.

Table 2. Description of the Symbol.

Notation	Definition
U_i	A user
CS	Contorl server
D_j	Drone
K_{global}	The global key generated by CS
ID_u	User's identity
ID_{cs}	CS's identity
ID_d	Drone's identity
PW_u	User's password
Bi	User's biometric
CB	Generating a cancelable biometric template
PID	Pseudo identity
C/R	PUF's Challenge-Response Pair
$E_K(\cdot)$	Symmetric encryption
$D_K(\cdot)$	Symmetric decryption
N	Random numbers
$\oplus$	Bitwise XOR operator
$\|$	Concatenation operator

4.1. Cancelable Biometric Template

Figure 2 illustrates the integration of the cancelable biometric mechanism with FE and PUF. The raw fingerprint vector is transformed into a cancelable version $\mathbf{y} = T_K(\mathbf{x})$, which is further processed through $FE.Gen(\cdot)$ to obtain auxiliary data σ, and used as input to the PUF for generating the response R. We propose an FFT-GRP-based method to improve efficiency and robustness.

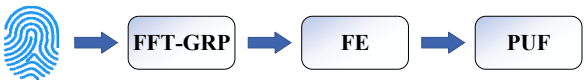

Figure 2. The cancelable method is combined with FE and PUF.

The transformation parameters (non-repeating random permutation array r, n $p \times q$-dimensional Gaussian Random Projection (GRP) matrices) adopt a design of dynamic generation and no local persistent storage to completely avoid the risk of parameter storage leakage. The specific implementation is as follows:

(1) Seed Generation: After the user enters the password PW_u, the local device invokes its own dedicated PUF, and combines the user-specific salt $Salt_u$ synchronized from the server, and generates a cryptographically secure seed through XOR operation and PUF mapping. The formula is $Seed = PUF_u(PW_u \oplus Salt_u)$. This seed integrates the secrecy of the password, the physical uniqueness of the PUF, and the randomness of the salt, and attackers cannot restore the seed with only a single element.

(2) Deterministic Parameter Generation: The $Seed$ is used as the initial input of the CSPRNG (Cryptographically secure pseudorandom number generator), ensuring the consistency and uniqueness of the generation results. Random permutation array r composed of non-repeating integers is generated iteratively, and elements following a Gaussian distribution with a mean of 0 and a variance of $1/p$ (p is the matrix dimension) are generated to construct n $p \times q$-dimensional GRP matrices.

(3) No-Storage Strategy: The generated r and GRP matrices only reside temporarily in the device memory to complete current FFT permutation, GRP transformation, and template matching operations. After the operations are completed, the temporary parameters in the memory are immediately destroyed, and no form of persistent storage (including plaintext, encrypted files, or hidden storage) is performed locally. The Control Server (CS) only stores the user's pseudo-identity PID_u, unique salt $Salt_u$, CRPs, and does not store any information directly related to transformation parameters, further reducing the risk of centralized leakage.

4.2. FFT-GRP

Although IoM is classic and efficient, Ghammam et al. [25] pointed out that IoM is not as resistant to various attacks as initially claimed, and the scheme is very susceptible to authentication and linking attacks. GRP-based IoM hashing [10] embeds the fingerprint vector into an n-dimensional Gaussian random subspace and extracts the index of the maximum projection feature. From the perspective of Locality Sensitive Hashing (LSH), this is equivalent to a hashing entry in the rank space [26]. The FFT-GRP method we propose combines GRP, Fast Fourier Transform, and random arrange to improve the efficiency and accuracy of fingerprint recognition.

GRP-based IoM hashing embeds the fingerprint vector into an n-dimensional Gaussian random subspace and extracts the index of the maximum projection feature. From the perspective of Locality Sensitive Hashing (LSH), this is equivalent to a hashing entry in the rank space [25]. Our method adopts this concept to enhance the efficiency and accuracy of fingerprint recognition. The method we use to generate templates is based on GRP and combines it with the Fast Fourier Transform (FFT) and random arrangement, which we call FFT-GRP. The specific process is as Figure 3 follows:

(1) FFT transformation is performed on the original fingerprint x (length is 299 bits) to generate complex vectors and sum the actual part and imaginary part to obtain a new real-value string $\bar{x}$ (length still remains 299 bits).

(2) Rearrange vector $\bar{x}$ using a non-repeating random array r, where $max(r) = length(r) = length(\bar{x})$. For each position i, set $\hat{x}_i = \bar{x}_{r_i}$. For example, if $r_1 = 10$, place the 10th element of $\bar{x}$ at the first position. In step 2, blocks of the same color are mapped vertically, changing their position indices while retaining their original values.

(3) Generate n random Gaussian projection matrices of dimensions $p \times q$. To facilitate matrix multiplication, it is required that $p = length(\hat{x})$. Multiply $\hat{x}$ with each of the n matrices to obtain n vectors of dimensions $1 \times q$, denoted as $x_i'(i \in [1, n])$.

(4) For the obtained n vectors $x_i'(i \in [1, n])$, record the index positions of their maximum and minimum values to generate a positional index vector of length $2n$. In step 4, red blocks represent the maximum value blocks, and blue blocks represent the minimum value blocks; their position indices are mapped to the new sequence.

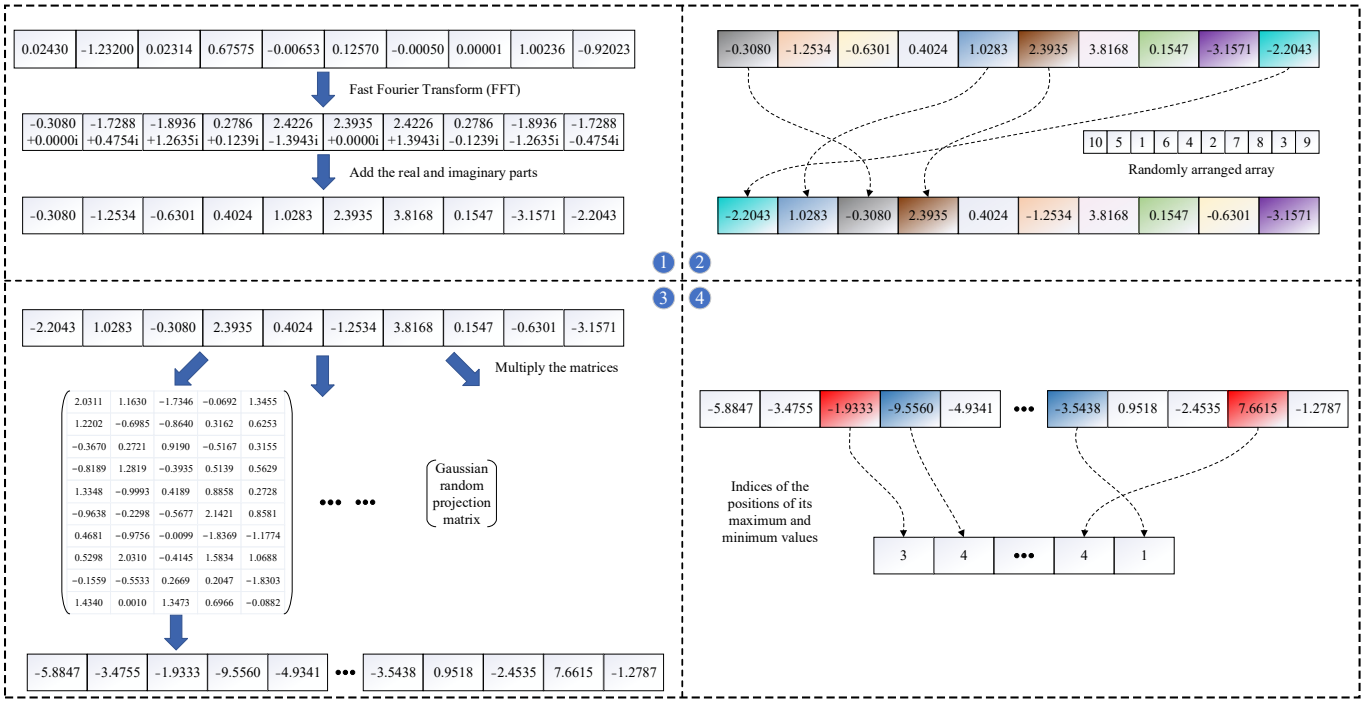

Figure 3. The process of generating cancelable biometric templates.

In our scheme, we first use the Fast Fourier Transform (FFT) to process the raw vector, converting fingerprint data from the spatial domain to the frequency domain. This makes the frequency characteristics of the fingerprint more pronounced, which is beneficial for subsequent processing. However, in practical applications, FFT is generally considered reversible. We then introduce random permutation and Gaussian Random Projections (GRP).

Unlike typical random projections, Gaussian random projection (GRP) matrices preserve the statistical structure and orthogonality of the original data. They enhance dimensionality reduction while maintaining discriminative features. The hash indices obtained from GRP also preserve Euclidean distance relations, which benefits template stability and matching performance [10].

4.3. User Registration Phase

In this phase, the user registers with the control server and obtains essential parameters for subsequent authentication.

Step-1: The user sends a registration request Reg_{req} to CS.

Step-2: Upon receiving the request, CS generates a challenge C_u and global key K_{global}, then returns C_u to the user.

Step-3: The user's device computes $R_u = PUF_u(C_u)$, inputs ID_u, PW_u, and biometric Bi_u. Using the cancelable transformation (Figure 3), it computes $Bi_u^* = CB(Bi_u)$ and derives $(\sigma_i, \tau_i) = Gen(Bi_u^*)$. Then, it computes $R_{\sigma_i} = PUF(\sigma_i)$ and $RPW = h(ID_u||PW_u||R_{\sigma_i})$. The tuple (R_u, ID_u, RPW) is sent to CS.

Step-4: CS computes a pseudo-identity $PID_u = ID_u \oplus h(K_{global}||ID_{cs})$ and $= PID_u \oplus h(ID_u||RPW)$. It stores $(PID_u, (C_u, R_u))$ for future authentication and replies with (ID_{cs}, A).

Step-5: The user calculates $ID'_{cs} = ID_{cs} \oplus R_{\sigma_i}$ and stores $(A, ID'_{cs}, RPW, \tau_i)$ in the smart card (SC).

4.4. Drones Registration Phase

Before deploying a new drone, you need to register with CS, and the specific steps are as follows.

Step-1: The drone sends a registration request to CS, where Reg_{req} represents the registration request.

Step-2: CS first generates an ID for the drone, denoted as ID_d, and a challenge C_d for the authentication process. CS calculates a pseudo identity for the drone $PID_d = ID_d \oplus h(K_{global}||ID_{cs})$. The data (PID_d, C_d) is sent to the drone.

Step-3: The drone uses C_d as input for its PUF and obtains the response $R_d = PUF_d(C_d)$. The drone stores PID_d in its device and sends R_d back to CS.

Step-4: CS stores the data $(ID_d, PID_d, (C_d, R_d))$ in the server.

4.5. Authentication Phase

After registration, the user uses the smart card to authenticate with the control server and drone. Figure 4 summarizes the protocol.

Step-1: The user inserts the smart card, inputs (ID_u, PW_u), and captures biometric Bi_u. The device computes the cancelable template $Bi_u^* = CB(Bi_u)$, then recovers $\sigma_i^* = Rep(\tau_i, Bi_u^*)$ and obtains $R_{\sigma_i^*} = PUF(\sigma_i^*)$. It computes RPW^* and compares with stored RPW. If matched, the user calculates PID_u^* and ID_{cs}^*, then generates timestamp t_1, and computes $M_1 = \mathrm{PID_u^*} \oplus ID_u$ and $M_2 = h(M_1||ID_u||ID_{cs}^*||t_1)$. The message (M_1, M_2, t_1) is sent to CS via a public channel.

Step-2: CS checks if t_1 is fresh. If valid, it uses stored PID_u to recover $ID_u = PID_u \oplus M_1$ and verifies M_2. Then, it selects (C_u, R_u) based on ID_u, generates t_2, computes TC_u and M_3, and sends (M_3, TC_u, t_2) to the user.

Step-3: The user validates t_2, derives C_u=$TC_u \oplus t_2$, and obtains $R_u = PUF_u(C_u)$. If matched with the original, it verifies M_3, then generates t_3 and random N_a, and computes $C_u^{new} = h(C_u||N_a||t_3) \oplus PID_u^*, R_u^{new} = PUF_u(C_u^{new})$. The user encrypts $M_4 = E_{PID_u^*}(N_b, R_u^{new})$, calculates $M_5 = h(R_u^{new}||N_a||ID_{cs}^*||t_3)$, and sends (M_4, M_5, t_3) to CS.

Step-4: CS checks t_3, decrypts M using PID_u to recover (N_a, R_u^{new}), and verifies M_5. If valid, it generates t_4 and random N_b, selects (C_d, R_d), and computes $TC_d = C_d \oplus h(N_b||t_4)$, $M_6 = E_{PID_d}(N_b, ID_d, PID_u, h(N_a||N_b))$, and $M_7 = h(R_d||h(N_a||N_b)||N_b||t_4)$. Then (M_6, M_7, TC_d, t_4) is sent to the drone.

Step-5: The drone validates t_4, decrypts M_6, derives C_d, and obtains $R_d = PUF_d(C_d)$. After verifying M_7, it generates t_5 and random N_c, computes $C_d^{new} = h(C_d||N_c||t_5) \oplus PID_d$, and $R_d^{new} = PUF_d(C_d^{new})$. It then calculates the session key $SK_{d2u} = h(PID_u||PID_d||h(N_a||N_b)||N_c)$, encrypts $M_9 = E_{PID_d}(N_c, R_d^{new})$, and computes M_8 and M_{10}. The tuple (M_8, M_9, M_{10}, t_5) is sent to CS.

Step-6: CS checks t_5, decrypts M_9 to get (N_c, R_d^{new}), and verifies M_{10}. It then computes C_u^{new} and C_d^{new}, updating the stored (C_u, R_u) and (C_d, R_d) to their new values. It generates t_6, prepares $M_{11} = E_{PID_u}(PID_d, N_b, N_c)$, $M_{12} = h(M_8||N_a||N_b||N_c||ID_u||ID_d||t_6)$, then sends (M_{11}, M_{12}, t_6) to the user.

Step-7: The user decrypts M_{11} to retrieve (PID_d, N_b, N_c) and calculates $SK_{u2d} = h(PID_u^*||PID_d||h(N_a||N_b)||N_c)$ and $ID_d = PID_d \oplus (PID_u^* \oplus ID_u)$. If M_8 and M_{12} are both valid, then all parties have authenticated each other and agreed on the session key $SK_{u2d} = SK_{d2u}$.

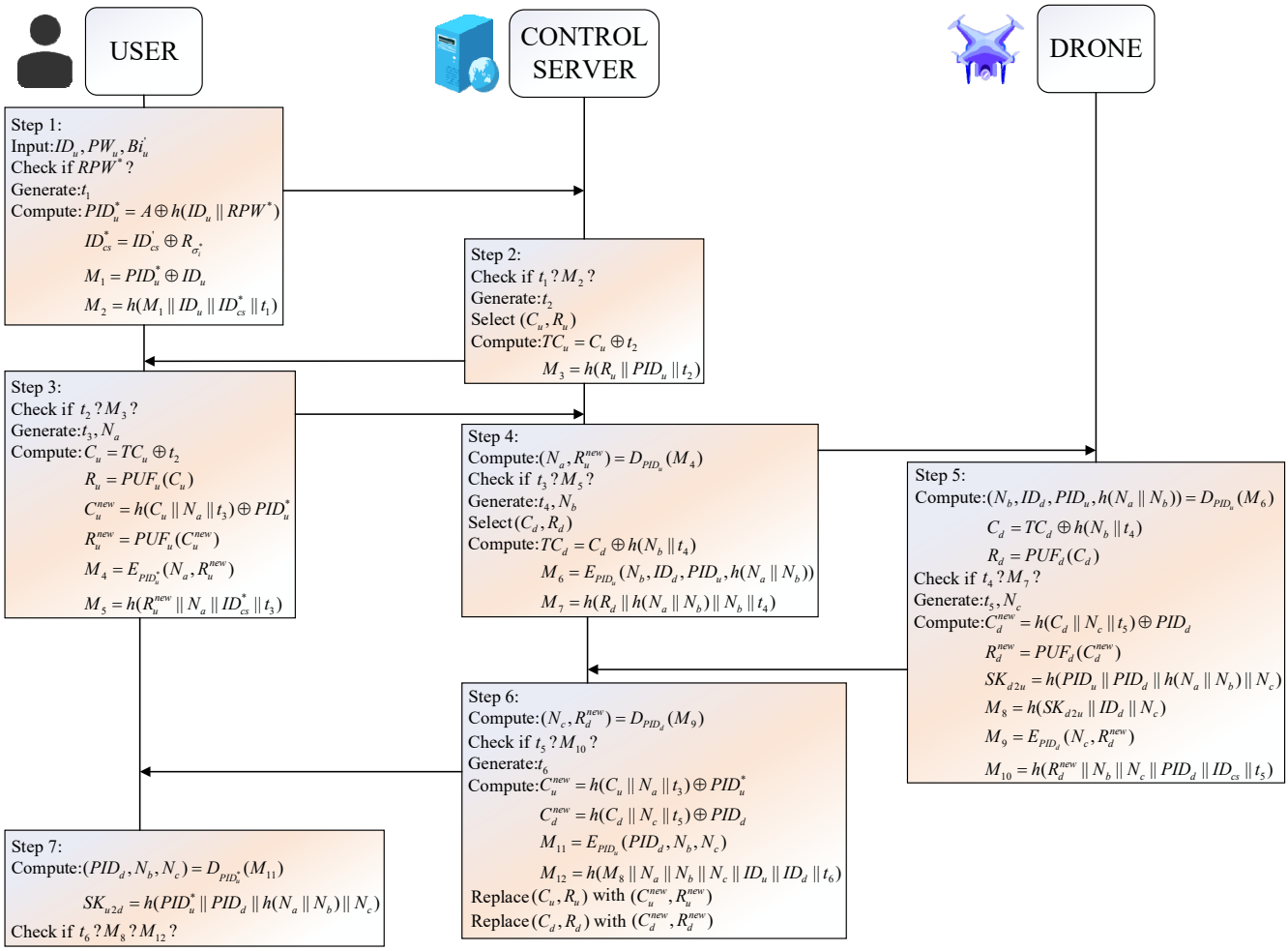

Figure 4. Authentication and key agreement phases for the proposed scheme.

4.6. Password/Biometrics Update Phase

To address template leakage or proactive privacy protection needs, this scheme designs a lightweight cancelable process based on the dynamic parameter generation mechanism. Template invalidation can be achieved only by updating the password or salt without modifying the user's original biometrics. The specific steps are as follows:

(1) The user enters a valid ID_u, PW_u, Bi_u', and calculates $Bi_u'' = CB(Bi_u')$; $\sigma_i^* = Rep(Bi_u'', \tau_i)$; $RPW^* = h(ID_u||PW_u||\sigma_i^*)$. If the calculated RPW^* does not match the RPW stored in the SC, the process is terminated. If they are the same, the user sends a template revocation request to the CS, including their own pseudo-identity PID_u and a fresh timestamp to prevent request replay attacks.

(2) After receiving the request, the CS verifies the validity of PID_u through pre-stored user authentication information and checks the freshness of the timestamp. If the verification passes, the user is allowed to perform the update operation.

(3) The user can choose two update methods: ① Password update: Enter a new password PW_u^{new} to generate a new seed $Seed^{new} = PUF_u(PW_u^{new} \oplus Salt_u)$; ② Salt update with retained password: The CS generates a new salt $Salt_u^{new}$ and synchronizes it to the user's device, generating a new seed $Seed^{new} = PUF_u(PW_u \oplus Salt_u^{new})$. The new seed generates a brand-new permutation array and GRP matrix through CSPRNG, and a new cancelable biometric template is generated based on the new parameters.

(4) Template Synchronization and Old Template Invalidation: The user's device synchronizes the CRP corresponding to the new template to the CS. The CS updates the stored user authentication information,

associates PID_u with the new template CRP, and permanently invalidates the old template corresponding to the old password/old salt. The old template cannot be restored through the new parameters, and even if leaked, it cannot be used for subsequent authentication, achieving efficient and secure revocability.

4.7. Drones Performs Self-Checks/Forms a Swarm

The process uses PUF (Physical Unclonable Function) hardware uniqueness, random number challenges, and hash verification to confirm the drone's legitimacy. Drones perform a self-check either on a schedule or before joining the swarm with the server. The following is the self-check process.

(1) CS generates a random number N_a and retrieves the drone's PID_d and initial challenge C_d. Calculate encrypted data $PID_d^* = PID_d \oplus N_a, N_a^* = N_a \oplus R_d$, R_d is initial PUF response, known only to CS and the drone. $M_1 = h(N_a||R_d||PID_d)$, CS sends the challenge message to the drone: (C_d, M_1, PID_d^*, N_a^*).

(2) The drone uses its PUF to compute $R_d' = PUF_d(C_d)$. Decrypt to get key data N_a and PID_d. Verify integrity $M_1' = h(N_a||R_d'||PID_d)$. If $M_1' \neq M_1$, reject the request. Generate drone-side random number N_b and calculate $(\sigma_i, \tau_i) = Gen(PID_d)$, $\tau_i' = N_b \oplus \tau_i$, $N_b^* = N_b \oplus R_d'$, $M_2 = h(\sigma_i||N_b)$. The drone sends the response message (τ_i', N_b^*, M_2) to CS.

(3) CS decrypts $N_b = N_b^* \oplus R_d$, $\tau_i = N_b \oplus \tau_i'$. Verify key σ_i: $\sigma_i^* = Rep(\tau_i, PID_d)$. Check if M_2, if verification passes, update the drone's pseudo-identity PID_d^{new}. Generate a cluster session key: SK = h($\tau_i||N_a||N_b||R_d$) for future communication. If verification fails, add the drone to the blacklist and block network access.

The core logic of drone swarm hierarchical authentication is that users only complete mutual authentication with the swarm Leader once, and ordinary drones, after passing self-check, access the swarm via "Control Server (CS) authentication + Leader authentication" without interacting with users to reduce redundant overhead, thereby reducing redundant overhead and effectively intercepting malicious drones.

(1) Ordinary drone sends a join request: Sends $Req_{join} = \{PID_d^{new}, t_{join}\}$ to CS; CS verifies timestamp freshness and whether PID_d^{new} is in the self-check-passed list.

(2) CS authenticates & issues a credential: CS generates temporary challenge C_{temp} and access credential $\text{Cert} = \text{h}(PID_{leader}||PID_d^{new}||K_{global})$, sends $M_1 = (PID_{leader}, C_{temp}, \text{Cert}, t_{cs})$ to ordinary drone; ordinary drone verifies t_{cs} and Cert integrity.

(3) Leader authenticates & grants access: Ordinary drone sends $M_2 = (PID_d^{new}, R_{temp}, \text{Cert}, t_{rd}), R_{temp} = \text{PUF}_d(C_{temp})$ to leader; leader verifies t_{rd} and Cert, synchronizes (C_{temp}, R_{temp}) to CS for PUF validity check; if passed, Leader generates cluster session key SK_{group}, encrypts and sends it, and synchronizes CS to update the swarm list.

5. Security Analysis of the Proposed Scheme

This section presents both formal and informal analyses, evaluating potential attacks using the ROR model and AVISPA tool.

5.1. Formal Analysis Under Real-or-Random (ROR) Model

In this section, we will introduce the use of the widely used ROR model [21] to prove the semantic security of the protocol. Table 3 shows the definition of the symbols and their query bounds.

Table 3. Description of the symbol.

Parameters	Definition	Query Bounds
$\mathcal{A}$	Polynomial-time adversary attacking Protocol $\mathcal{P}$ in the random oracle model	--
l	Bit length of the biometric feature Bi_i	Corresponding to the dimension of the fingerprint feature vector (Section 4.1)
q_h	Upper bound of Hash Oracle queries.	Covers all calls to $h(\,)$ (e.g., RPW, $M_2/M_3/M_5/PID_u/PID_d$/SK) generation; $q_h \leq poly(n)$
q_P	Upper bound of PUF Oracle queries.	Covers $PUF_u(\,)$ (generating R_u, R_u^{new}) and $PUF_d(\,)$ (generating R_d, R_d^{new}); $q_P \leq poly(n)$
q_s	Upper bound of Send Oracle queries	Covers interception/forgery/modification of public-channel messages; $q_s \leq poly(n)$
$\|RH\|$	Size of the hash function output domain	Determined by hash output bit length (e.g., 256 bits for SHA-256)
$\|RP\|$	Size of the PUF response domain	Determined by PUF hardware implementation in resource-constrained devices
$\|DC\|$	Size of the password dictionary for guessing attacks	Consistent with the Adversary Model (Section 2.2)

Security Proof: The security proof provided in Theorem 1 is similar to the security proof given in [21,22].

Theorem 1. *Let $\mathcal{A}$ be a polynomial-time adversary attacking protocol $\mathcal{P}$ in the random oracle model. Table 3 definitions of parameters and query bounds. Then:*

$$Adv_p(A) \leq \frac{q_h^2}{|RH|} + \frac{q_P^2}{|RP|} + \frac{q_s}{2^{l-1} \cdot |DC|} \tag{1}$$

Proof. Five games, denoted as $G_i (i = 0, 1, 2, 3, 4)$ are defined. Let $Succ_i$ represent the event that the attacker $\mathcal{A}$ successfully guesses the bit c in game G_i. □

G_0: Fully simulates the real execution of Protocol $\mathcal{P}$. Hash operations, PUF responses, fuzzy extractor (FE) operations, and timestamp verification are performed in accordance with the original protocol logic. The adversary's advantage is defined as:

$$adv_{\mathcal{P}}(\mathcal{A}) = |2 \cdot Pr[Succ_0] - 1| \tag{2}$$

G_1: In G_1, the attacker performs an eavesdropping attack on $\mathcal{P}$. $\mathcal{A}$ can intercept authentication phase messages like (M_1, M_2, t_1), *etc*. In this game, $\mathcal{A}$ makes an $Execute(\Pi^t, \Pi^u)$ (intercept public channel messages such as (M_1, M_2, t_1)) and a *Test* query (verify whether the session key is random) to verify whether it is the true session key *SK* or a random number. The session key *SK* is calculated as: $SK = h(ID_u||PID_{RD}||ID_{RD}||h(N_b||N_c)||N_d)$. But this SK isn't on the public channel, and relies on random numbers and encrypted pseudo-identities ($PID_u = ID_u \oplus h(K_{global}||ID_{cs})$). So $\mathcal{A}$ can't get or derive SK. Associated Components: One-wayness of the hash function $h(\cdot)$, encrypted pseudo-identity mechanism. Therefore, the probability of $\mathcal{A}$ winning G_1 by eavesdropping on the exchange messages does not increase. That is to say,

$$Pr[Succ_1] = Pr[Succ_0] \tag{3}$$

G_2: On the basis of G_1, simulate Hash Oracle responses—all $h(\cdot)$ queries from the adversary return random values conforming to the hash distribution. Transition Argument: According to the birthday paradox, the upper bound of the hash collision probability is $\frac{q_h^2}{2|RH|}$. Associated Components: Random oracle

assumption of the hash function (used to generate RPW, M_2, M_3, *etc.*). The random oracle property makes the simulated responses indistinguishable from real responses, as:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2|RH|} \tag{4}$$

G_3: On the basis of G_2, simulate PUF Oracle responses-all PUF(·) queries from the adversary return random values conforming to the PUF distribution. But PUF has physical unclonability and pseudorandomness. Associated Components: PRF property of PUF (generating challenge-response pairs C/R). and its responses are indistinguishable from random numbers. The upper bound of the collision probability is $\frac{q_P^2}{2|RP|}$, as:

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{q_P^2}{2|RP|} \tag{5}$$

G_4: In this game model, no random oracle is assumed. The probability of $\mathcal{A}$ successfully forging all messages depends on the inclusion of *CorruptDevice* operations. In this scenario, $\mathcal{A}$ can retrieve all secret data stored in the user's device $(A, ID_{CS}, RPW, \tau_i)$. However, $\mathcal{A}$ cannot derive any valuable information from this data, as the user's *ID*, password, and biometric features are not directly stored on the smart card. Instead, the card stores data computed from these elements. $RPW = h(ID_u||PW_u||R_{\sigma_i})$ relies on R_{σ_i} generated by FE and PUF. Assuming the scheme replaces the original mechanism with k random bits, the probability of guessing the biometric password approximates $\frac{1}{2^k}$. The adversary needs to guess both the password (from DC) and the biometric (length k), with a success probability upper bound of $\frac{q_s}{2^k \cdot |DC|}$.The system also limits the number of incorrect password attempts. Since games G_3 and G_4 remain identical under scenarios without guessing attacks, we derive:

$$|Pr[Succ_4] - Pr[Succ_3]| \leq \frac{q_s}{2^k \cdot |DC|} \tag{6}$$

Note that the session keys for all random oracles are simulated in game G_4. So after the *Test* query is completed, $\mathcal{A}$ needs to guess bit c to win the game. We infer that:

$$Pr[Succ_4] = \frac{1}{2} \tag{7}$$

Based on (1), (2), and (6), we can first derive the following relationship:

$$\frac{1}{2} adv_{\mathcal{P}}(\mathcal{A}) = |Pr[Succ_0] - \frac{1}{2}| = |Pr[Succ_1] - \frac{1}{2}| = |Pr[Succ_1] - Pr[Succ_4]| \tag{8}$$

Using the triangular inequality and (6), we further derive:

$$\begin{aligned} |Pr[Succ_1] - \frac{1}{2}| &= |Pr[Succ_1] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_3]| + |Pr[Succ_3] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_3]| \\ &\quad + |Pr[Succ_3] - Pr[Succ_4]| \end{aligned} \tag{9}$$

Based on (3), (4), and (5), we can deduce:

$$|Pr[Succ_1] - \frac{1}{2}| = |Pr[Succ_1] - Pr[Succ_4]| \leq \frac{q_h^2}{2|RH|} + \frac{q_P^2}{2|RP|} + \frac{q_s}{2^k \cdot |DC|} \tag{10}$$

Thus:

$$\frac{1}{2}adv_{\mathcal{P}}(\mathcal{A}) \leq \frac{q_h^2}{2|RH|} + \frac{q_P^2}{2|RP|} + \frac{q_s}{2^k \cdot |DC|} \tag{11}$$

In conclusion, this result demonstrates that the proposed scheme achieves session key security:

$$adv_{\mathcal{P}}(\mathcal{A}) \leq \frac{q_h^2}{|RH|} + \frac{q_P^2}{|RP|} + \frac{q_s}{2^{l-1} \cdot |DC|} \tag{12}$$

The security of the protocol relies on the synergy of four core components, which support the corresponding game transitions respectively:

(1) Hash Function $h(\cdot)$: Acts as a random oracle to generate message authentication codes, pseudo-identities, and session keys, supporting the transition from G_1 to G_2 and resisting hash collision attacks;
(2) PUF/PRF: The physical unclonability and pseudorandomness of PUF ensure the security of C/R pairs, supporting the transition from G_2 to G_3 and resisting PUF response forgery attacks;
(3) FE/Rep: The noise resistance of the Fuzzy Extractor ensures the matching accuracy of biometric templates. Combined with PUF, it generates R_{σ_i}, supporting the transition from G_3 to G_4 and resisting biometric guessing attacks;
(4) Timestamps $t_1 - t_6$: Prevent replay attacks through freshness verification, avoid the adversary interfering with game execution, and ensure the validity of all transitions.

5.2. Formal Analysis under Avispa Tool

We use the AVISPA tool [27] to formally verify the security of the proposed protocol. AVISPA simulates against active threats, including replay and man-in-the-middle attacks, using the Dolev–Yao (DY) model [23]. The protocol is first written in High-Level Protocol Specification Language (HLPSL), then translated to Intermediate Format (IF) via the HLPSL2IF converter, and finally analyzed using AVISPA's backends. The Output Format (OF) clearly states whether an attack is possible. The intruder model actively participates in sessions. As shown in Figure 5 simulation under the SPAN interface confirms that the proposed protocol resists both replay and man-in-the-middle attacks.

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/Formal Security Analysis Based on AVISPA tool.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed  : 9 states
 Reachable : 4 states
```

Figure 5. Analysis of simulation results under CL-AtSe and OFMC backends.

5.3. Informal Analysis

We believe users' mobile devices and drones are vulnerable, with stored data at risk of theft. Data is transmitted via public channels, making it easy to intercept. However, we consider this protocol safe and capable of achieving the following objectives. Table 4 shows the comparison of the proposed scheme with

other schemes in the field in terms of their resistance to different attacks. Where " √ " indicates that it can be done, " × " indicates that it cannot be done, and "NA" indicates discussion or ambiguity.Each claimed security property is mapped to the adversary's capabilities (from Section 2.2) and the corresponding protocol steps that ensure the property, as follows:

(1) A1 (User Anonymity):

The corresponding threat is ID disclosure via public channel interception, and the adversary's capability is II.B.1 (public channel interception). The protocol ensures this property through the following steps: During user registration, $PID_u = ID_u \oplus h(K_{global}||ID_{cs})$ is executed (Step 4); during authentication, the real ID is hidden via $M_1 = \mathrm{PID}_u^* \oplus ID_u$ (Step 1), and only the Control Server (CS) can recover ID_u in Step 2, preventing ID leakage over public channels.

(2) A2 (Forward Secrecy):

The corresponding threat is the compromise of long-term keys/CRPs, and the adversary's capabilities are II.B.2 (drone memory compromise) and II.B.3 (partial parameter leakage). The protocol's guarantee steps include: After authentication, CRP updates are performed, *i.e.*, $C_u^{new} = h(C_u||N_a||t_3) \oplus PID_u^*$ and $C_d^{new} = h(C_d||N_c||t_5) \oplus PID_d$ (Steps 3, 5), and the Control Server updates the stored (C_u, R_u) and (C_d, R_d) in Step 6, ensuring that even if long-term keys or old CRPs are compromised, past session keys remain secure.

(3) A3 (Sensor Anonymity):

The corresponding threat is drone ID disclosure, and the adversary's capability is II.B.1 (public channel interception). The protocol generates a pseudo-identity via $PID_d = ID_d \oplus h(K_{global}||ID_{cs})$ during drone registration (Step 2) and hides ID_d in the encrypted message M_6 during authentication (Step 4), preventing the drone's real ID from being intercepted and obtained.

(4) A4 (Replay Attack Resistance):

The corresponding threat is the replay of authentication messages (M_1–M_{12}), and the adversary's capability is II.B.1 (message replay). The protocol performs freshness checks on timestamps t_1–t_6 (Steps 2, 3, 4, 5, 6) and introduces random numbers $N_a/N_b/N_c$ into session key calculation (Steps 3, 4, 5), ensuring that replayed old messages are rejected due to invalid timestamps or mismatched random numbers.

(5) A5 (Man-in-the-Middle Attack Resistance):

The corresponding threat is message tampering/injection, and the adversary's capability is II.B.1 (message modification/injection). The protocol verifies message integrity through a hash verification mechanism (e.g., $M_2 = h(M_1||ID_u||ID_{cs}^*||t_1)$ (Step 1), $M_3 = h(R_u||PID_u||t_2)$ (Step 2), *etc*.) and performs symmetric encryption on sensitive data $M_4 = E_{PID_u^*}(N_b, R_u^{new})$, $M_6 = E_{PID_d}(N_b, ID_d, PID_u, h(N_a||N_b))$ to resist man-in-the-middle attacks on message tampering and injection.

(6) A6 (Physical Attack Resistance):

The corresponding threat is Physical Unclonable Function (PUF) tampering/physical cloning, and the adversary's capability is II.B.2 (drone memory compromise). The protocol leverages the unclonability of PUF—Challenge-Response Pairs (*C*,*R*) are bound to physical devices, and tampering invalidates PUF responses (Section 3.3)—while decoupling templates from physical biometrics via cancelable biometric transformation (FFT-GRP) (Section 4.2) to resist physical attacks on PUF and biometrics.

(7) A7 (Device Loss Resilience):

The corresponding threat is smart card/drone loss with data extraction, and the adversary's capability is II.B.2 (smart card/drone data extraction). The protocol ensures that smart cards only store derived data $RPW = h(ID_u||PW_u||R_{\sigma_i})$ instead of raw IDs, passwords, or biometrics (Step 3), and drones only store

pseudo-identity PID_d without plaintext ID_d or global key K_{global} (Step 3), preventing sensitive information from being directly obtained if the device is lost.

(8) A8 (Known Session Key Attack Resistance):

The corresponding threat is session key leakage leading to the compromise of past/future keys, and the adversary's capability is II.B.1 (message interception). The protocol's guarantee mechanisms include: The session key $SK = h(PID_u^*||PID_d||h(N_a||N_b)||N_c)$ (Steps 5, 7) uses unique random numbers for each session, and Challenge-Response Pairs are updated after each authentication (Steps 3, 5, 6), ensuring that the leakage of a single session key does not affect the security of other sessions.

(9) A9 (Password Guessing Attack Resistance):

The corresponding threat is online/offline password guessing via a dictionary, and the adversary's capability is II.B.4 (guessing attacks). The protocol combines the password with biometric-derived R_{σ_i} to generate RPW ($RPW = h(ID_u||PW_u||R_{\sigma_i})$, Step 3) and limits the number of incorrect RPW matching attempts through the system (Section 4.6), significantly reducing the probability of successful password guessing.

(10) A10 (User Device Loss Attack Resistance):

The corresponding threat is stolen devices leading to identity impersonation, and the adversary's capability is II.B.2 (smart card data extraction). The protocol requires biometric verification during authentication, *i.e.*, recovering σ_i^* via $Bi_u^* = CB(Bi_u)$ and $\sigma_i^* = Rep(\tau_i, Bi_u^*)$ (Step 1), and the calculation of PID_u and ID_{cs} depends on valid RPW matching (Step 1). Even if smart card data is extracted, the adversary cannot complete impersonation due to the lack of biometrics.

(11) A11 (Offline Attack Resistance):

The corresponding threat is offline brute-force attacks on intercepted encrypted messages, and the adversary's capability is II.B.1 (message interception). The protocol uses pseudo-identities (PID_u, PID_d) as keys for encrypted messages (M_4, M_6, M_9, M_{11}) (Steps 3, 4, 5, 6), and pseudo-identities are derived from the global keyK_{global} unknown to the adversary, making offline cracking difficult without valid keys.

(12) A12 (Sensor Theft Attack Resistance):

The corresponding threat is biometric template leakage due to stolen biometric sensors, and the adversary's capability is II.B.2 (sensor data extraction). The protocol generates cancelable biometric templates (CB) via FFT-GRP transformation (Section 4.2), making it impossible to reverse the original biometric Bi_u from CB, and templates can be updated by modifying random permutation arrays r or projection matrices, avoiding permanent biometric leakage after sensor theft.

(13) A13 (Biometric Attack Resistance):

The corresponding threat is biometric template inversion/replay, and the adversary's capability is II.B.3 (partial parameter leakage). The protocol prevents template inversion via $CB = FFT - GRP(Bi_u) + PUF(\sigma_i^*)$ (Section 4.1), and the Rep function of the fuzzy extractor requires Bi_u to be sufficiently similar to the registered Bi_u ($dis(Bi, Bi_u) < t$, Section 4.1), resisting replayed forged biometric data.

(14) A14 (Password/Biometric Update Support):

The corresponding threat is the inability to update credentials after compromise, and the adversary's capability is II.B.2 (device data extraction). During the update phase, the protocol allows users to enter a new password (PW_u^{new}) or new biometric (Bi_u^{new}), calculate a new $RPW^* = h(ID_u||PW_u||\sigma_i^*)$ and replace the old RPW (Steps 1–2), enabling secure credential updates.

(15) A15 (Mutual Authentication):

The corresponding threat is unilateral authentication failure, and the adversary's capability is II.B.1 (impersonation via message forgery). The protocol ensures mutual authentication through three-level verification: Between the user and the Control Server via M_2 (user to CS) and M_3 (CS to user) (Steps 1–2),

between the Control Server and the drone via M_7 (CS to drone) and M_8/M_{10} (drone to CS) (Steps 4–5), and between the user and the drone via M_{12} (CS to user) and session key consistency check (Step 7), ensuring that the identities of all three parties are legally verified.

(16) A16 (PUF Update Support):

The corresponding threat is stale Challenge-Response Pairs leading to PUF response forgery, and the adversary's capability is II.B.2 (drone memory compromise). The protocol implements PUF updates by generating new Challenge-Response Pairs, *i.e.*, $C_u^{new} = h(C_u||N_a||t_3) \oplus PID_u^*, R_u^{new} = PUF_u(C_u^{new})$ (Step 3), and $C_d^{new} = h(C_d||N_c||t_5) \oplus PID_d, R_d^{new} = PUF_d(C_d^{new})$ (Step 5), and the control server updates the stored (C_u, R_u) and (C_d, R_d) in Step 6, avoiding PUF response forgery using old CRPs.

(17) A17 (Avispa-Verified Security):

The corresponding threats are replay/man-in-the-middle attacks, and the adversary's capability is II.B.1 (message replay/injection). The protocol undergoes Avispa simulation via HLPSL specification (Section 5.2), and tool verification confirms its effective resistance to the aforementioned threats, ensuring the protocol's security in simulated attack scenarios.

Table 4. Comparison of the Security Properties.

Scheme	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17
[13]	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
[17]	√	√	NA	√	√	√	√	×	√	√	√	√	NA	√	√	NA	NA
[18]	√	√	√	√	√	√	√	×	√	√	√	√	√	×	√	√	√
[19]	√	√	√	√	√	√	√	NA	√	√	√	√	√	×	√	×	NA
Ours	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

A1: User Anonymity; A2: Forward Secrecy; A3: Sensor Anonymity; A4: Replay Attack; A5: Man-in-the-Middle Attack; A6: Physical Attack; A7: Device Loss Attack; A8: Known session Key attack; A9: Password Guessing Attack; A10: User Device Loss Attack; A11: Offline Attack; A12: Sensor Theft Attack; A13: Biometric Attack; A14: Password/Biometric update; A15: Mutual authentication; A16: PUF update; A17: Avispa.

6. Performance Analysis and Comparison

6.1. Accuracy Performance Evaluation

The experiment uses four public fingerprint datasets: FVC2002 DB1, DB2, and FVC2004 DB1, DB2, each containing 800 images from 100 users with 8 samples per finger. Following Jin's method [28], get three samples are used for training and five for testing. The paper adopts the modified Polar Grid-based Three-Tuple Quantization (PGTQ) for feature extraction, which has an alignment-free property. Taking each minutia as a reference point, only neighboring minutiae within a 70-pixel radius are considered. After rotation-translation correction and polar coordinate conversion, they are quantized into triplets with steps of 10 pixels for radius, 20° for radial angle, and 30° for orientation angle. A polar grid containing more than one minutia is mapped to bit "1", and finally, a variable-length binary vector is formed. Matching adopts a two-stage strategy consisting of local descriptor intersection matching and global score ratio calculation. Then evaluates system performance using genuine/imposter scores and the Equal Error Rate (EER), as per [29].

For each dataset, on an experimental platform with hardware configured as Intel(R) Core(TM) i5-12400F with 16 GB of memory, and software as MATLAB 9.12.0.1884302 (R2022a) , randomly generate 200 passwords of varying lengths as seeds, run 100 times, and record their minimum, maximum, average, and variance values. Figure 6 presents the EER and ROC results on FVC2002 DB1 and FVC2004 DB1. The ROC curves for DB1 (2002) and DB1 (2004) are nearly ideal, with steep curves approaching the top-left, indicating

strong performance. As shown in Table 5, our method achieves the lowest EER on three datasets and performs competitively on the fourth. Compared to GRP hashing [10], our method significantly improves accuracy.

Table 5. EER Comparison between Proposed Method and Other Methods.

Methods		FVC2002-DB1	FVC2002-DB2	FVC2004-DB1	FVC2004-DB2
GRP-based IoM hashing [10]		0.22%	0.47%	4.74%	3.99%
URP-based IoM hashing [10]		0.43%	2.1%	4.51%	8.02%
WSE hash [30]		0.2%	0.62%	2.6%	7.13%
Bloom Filter [31]		2.3%	1.8%	13.4%	8.1%
Ours	Min	0.00%	0.07%	2.91%	3.60%
	Max	0.30%	0.55%	4.09%	4.84%
	Average	0.14%	0.29%	3.37%	4.39%
	Variance	0.0036	0.0054	0.0442	0.0528

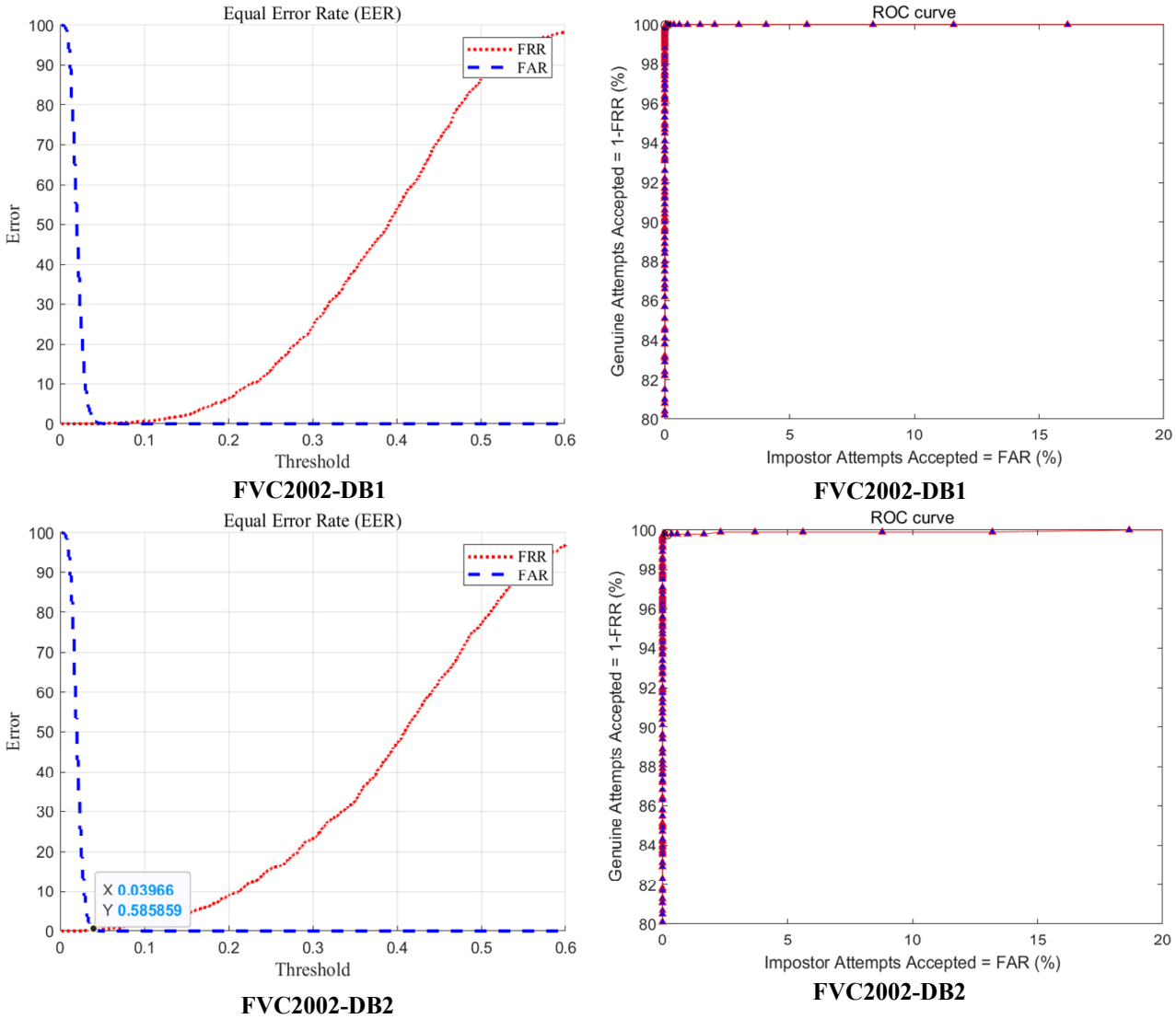

Figure 6. EER and ROC for FVC2002-DB1 (up two) and FVC2004-DB1 (down two).

6.2. Unlinkability

Biometric identification systems must satisfy unlinkability. This means that two or more repeatable templates generated from the same biometric features cannot be linked to each other, nor can it be inferred that they originate from the same biometric [4].

To verify that our proposed template protection scheme meets the requirement of unlinkability, we adopt a method proposed in [32] that uses a global metric $D_{\leftrightarrow}^{sys}$ to evaluate the unlinkability of the system. $D_{\leftrightarrow}^{sys}$ has values ranging from 0 to 1; $D_{\leftrightarrow}^{sys}$ value of 1 suggests that the system is fully linkable, while a value of 0 indicates complete unlinkability. Generate 10 random keys as parameters to perform 10 transformations on each fingerprint vector, resulting in a total of 5000 transformed templates; and then calculate two scores for evaluation, namely Mated score and Non-Mated score. Based on these two evaluation scores, $D_{\leftrightarrow}^{sys}$ is calculated. Figure 7 shows our assessment of the scheme using the FVC2002-DB1 and FVC2004-DB1 fingerprint databases.

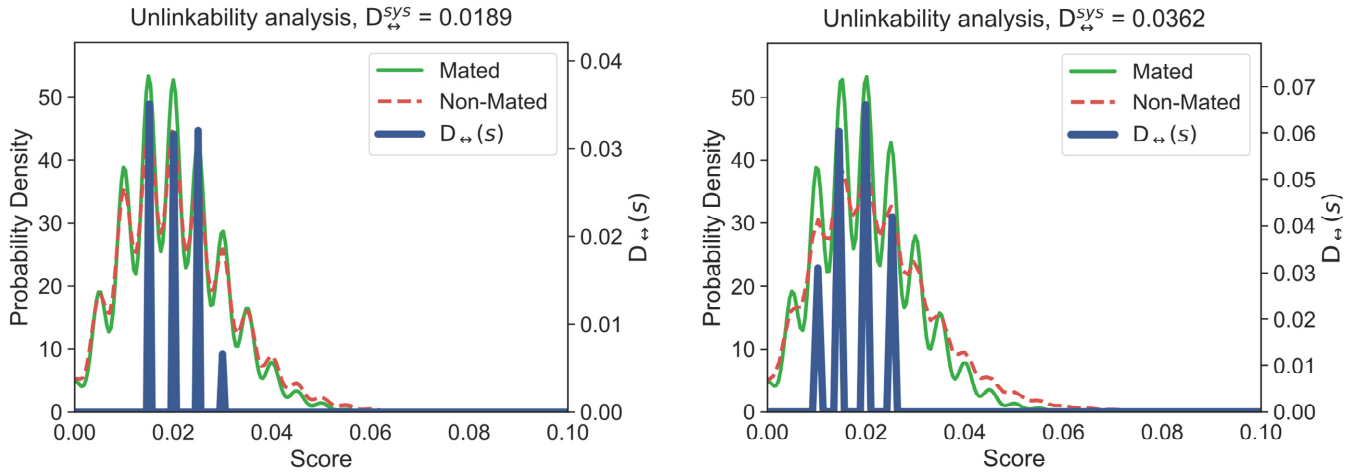

Figure 7. The unlinkability of FVC2002-DB1 (**left**) and FVC2004-DB1 (**right**).

6.3. Computation Cost

To intuitively analyze the performance of the proposed scheme, we compare its computational cost with that of existing protocols. Let T_h, T_{PUF}, T_{FE}, and $T_{E/D}$ denote the time required for hash operations, PUF evaluations, fuzzy extractor computation, and symmetric encryption/decryption (e.g., AES-128), respectively. Bitwise XOR operations are excluded due to negligible cost. Based on experimental results from [20] (The hardware configuration of this experiment is as follows: the user device adopts an HTC One smartphone, and the server uses an Intel Core i5-4300 machine; in terms of core technologies and tools, a 128-bit arbiter PUF is used for PUF operations, BCH code is adopted for the generation (FE.Gen) and reconstruction (FE.Rec) operations of the fuzzy extractor, and the JCE library is used to estimate the execution time of cryptographic operations) and [33] (The paper obtains the values of the operation times such as modular exponentiation and symmetric encryption/decryption through benchmark parameter setting and logical conversion. Instead of being derived from actual measurement through self-built hardware experiments, these values are based on standardized conversion supported by existing research conclusions in the field of cryptography), we adopt $T_h = 0.056$ ms, $T_{PUF} = 0.13$ ms, $T_{FE} = 3$ ms, and $T_{E/D} = 8.7$ ms. Accordingly, our mutual authentication phase completes in 74.408 ms: 21.164 ms for the user, 35.304 ms for the server, and 17.94 ms for the drone.

For comparison, other protocols incur costs from modular arithmetic and elliptic curve operations, with $T_m = 16.84$ ms, $T_{eca} = 4.4$ ms, and $T_{ecm} = 17.1$ ms [21]. As shown in Table 6, our scheme achieves lower computation overhead while maintaining strong security, unlike the overly lightweight design in [17], which compromises robustness.

Table 6. Computational Cost Comparisons.

Scheme	User-Side Computation Cost	Server-Side Computation Cost	Drone-Side Computation Cost	Total Cost
[1]	$16T_h + T_{FE} + 5T_{ecm} + 2T_{eca}$	$10T_h + 2T_{ecm} + T_{eca}$	$7T_h + 3T_{ecm} + T_{eca}$	193.448 ms
[16]	$15T_h + 2T_m$	$9T_h$	$7T_h$	35.416 ms
[17]	$8T_h + 2T_{E/D} + T_{FE}$	$11T_h + 6T_{E/D} + T_{FE}$	$6T_h + 2T_{E/D} + T_{FE} + 2T_{PUF}$	97.66 ms
[18]	$7T_h + 3T_{E/D} + T_{exp}$	$2T_h + 3T_{E/D} + 2T_{exp}$	$3T_h + 2T_{E/D} + 1T_{exp}$	171.312 ms
Ours	$9T_h + 2T_{PUF} + 2T_{E/D} + T_{EF}$	$9T_h + 4T_{E/D}$	$5T_h + 2T_{PUF} + 2T_{E/D}$	74.408 ms

7. Conclusions

This paper proposes a method that combines the Fast Four Transform, Gaussian Random Projections, Position-Sensitive Hashing, Fuzzy Extractors, and Physical Unclonable Function to generate cancelable biometric templates, addressing the issue of permanent loss of biometric templates, and enhancing the protection of sensitive biometric data while ensuring strong resistance to known attacks. Experimental evaluations on FVC2002 and FVC2004 datasets demonstrate high accuracy and unlinkability. The complete AKA protocol introduced in this paper incorporates cancelable methods and PUF update schemes, allowing the generation of new templates by modifying parameters in case of data theft, ensuring security and efficiency. Formal security analysis using the ROR model and AVISPA simulation confirms the protocol's robustness against replay, man-in-the-middle, and device-corruption attacks. Compared with existing protocols, our method achieves a better balance between security strength, computational efficiency, and biometric privacy, making it suitable for deployment in resource-constrained IoD environments.

Author Contributions

Conceptualization, K.C. and W.B.; Methodology, K.C.; Software, K.C.; Validation, K.C., W.B. and D.X.; Formal Analysis, K.C.; Investigation, Q.L.; Resources, W.B.; Data Curation, W.B.; Writing—Original Draft Preparation, K.C.; Writing—Review & Editing, W.B.; Visualization, J.M.; Supervision, W.B.; Project Administration, W.B.; Funding Acquisition, W.B. and D.X.

Ethics Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Bundled with the book Handbook of Fingerprint Recognition (3rd Ed., Springer, 2022). Available via a formal license application to the FVC organizers (University of Bologna).

Funding

The work are partially supported by the National Natural Science Foundation of China (No. 61801004), Natural Science Foundation of Anhui Province (No. 2108085MF206).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Mekdad Y, Aris A, Babun L, Fergougui AE, Conti M, Lazzeretti R, et al. A survey on security and privacy issues of uavs. *Comput. Netw.* **2023**, *224*, 109626. DOI:10.1016/j.comnet.2023.109626
2. Gharibi M, Boutaba R, Waslander SL. Internet of drones. *IEEE Access* **2016**, *4*, 1148–1162. DOI:10.1109/ACCESS.2016.2537208
3. Li B, Fei Z, Zhang Y. Uav communications for 5g and beyond: Recent advances and future trends. *IEEE Internet Things J.* **2019**, *6*, 2241–2263. DOI:10.1109/JIOT.2018.2887086
4. Patel VM, Ratha NK, Chellappa R. Cancelable biometrics: A review. *IEEE Signal Process. Mag.* **2015**, *32*, 54–65. DOI:10.1109/MSP.2015.2434151
5. Choudhury B, Then P, Issac B, Raman V, Haldar MK. A survey on bio metrics and cancelable biometrics systems. *Int. J. Image Graph.* **2018**, *18*, 1850006. DOI:10.1142/S0219467818500067
6. Wang D, Cheng H, Wang P, Huang X, Jian G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. DOI:10.1109/TIFS.2017.2721359
7. Manisha, Kumar N. Cancelable biometrics: A comprehensive survey. *Artif. Intell. Rev.* **2020**, *53*, 3403–3446. DOI:10.1007/s10462-019-09767-8
8. Teoh ABJ, Yuang CT. Cancelable biometrics realization with multispace ran dom projections. *IEEE Trans. Syst. Man Cybern. Part B* **2007**, *37*, 1096–1106. DOI:10.1109/TSMCB.2007.903538
9. Algarni AD, El Banby GM, Soliman NF, El-Samie FEA, Iliyasu AM. Efficient implementation of homomorphic and fuzzy transforms in random-projection encryption frameworks for cancellable face recognition. *Electronics* **2020**, *9*, 1046. DOI:10.3390/electronics9061046
10. Jin Z, Hwang JY, Lai YL, Kim S, Teoh ABJ. Ranking-based locality sen sitive hashing-enabled cancelable biometrics: Index-of-max hashing. *IEEE Trans Actions Inf. Forensics Secur.* **2018**, *13*, 393–407. DOI:10.1109/TIFS.2017.2753172
11. Kuzu RS, Piciucco E, Maiorana E, Campisi P. On-the-fly finger-vein-based biometric recognition using deep neural networks. *IEEE Trans. Inf. Tion Forensics Secur.* **2020**, *15*, 2641–2654. DOI:10.1109/TIFS.2020.2971144
12. Das ML. Two-factor user authentication in wireless sensor networks. *IEEE Trans Actions Wirel. Commun.* **2009**, *8*, 1086–1090. DOI:10.1109/TWC.2008.080128
13. Srinivas J, Das AK, Wazid M, Vasilakos AV. Designing secure user authentication protocol for big data collection in iot-based intelligent transportation system. *IEEE Internet Things J.* **2021**, *8*, 7727–7744. DOI:10.1109/JIOT.2020.3040938
14. Sarier ND. Practical multi-factor biometric remote authentication. In Proceedings of the 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington, DC, USA, 27–29 September **2010**; pp. 1–6.
15. Kirsal Ever Y. Secure-anonymous user authentication scheme for e-healthcare applica tion using wireless medical sensor networks. *IEEE Syst. J.* **2019**, *13*, 456–467. DOI:10.1109/JSYST.2018.2866067
16. Gope P, Das AK, Kumar N, Cheng Y. Lightweight and physically secureanonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4957–4968. DOI:10.1109/TII.2019.2895030
17. Chaudhary D, Soni T, Vasudev KL, Saleem K. A modified lightweight au thenticated key agreement protocol for internet of drones. *Internet Things* **2023**, *21*, 100669. DOI:10.1016/j.iot.2022.100669
18. Liu Z, Guo C, Wang B. A physically secure, lightweight three-factor and anony mous user authentication protocol for iot. *IEEE Access* **2020**, *8*, 195914–195928. DOI:10.1109/ACCESS.2020.3034219
19. Tanveer M, Khan AU, Kumar N, Hassan MM. Ramp-iod: A robust authenticated key management protocol for the internet of drones. *IEEE Internet Things J.* **2022**, *9*, 1339–1353. DOI:10.1109/JIOT.2021.3084946
20. Bian W, Gope P, Cheng Y, Li Q. Bio-aka: An efficient fingerprint based two factor user authentication and key agreement scheme. *Future Gener. Comput. Syst.* **2020**, *109*, 45–55. DOI:10.1016/j.future.2020.03.034
21. Zhang H, Bian W, Jie B, Xu D, Zhao J. A complete user authentication and key agreement scheme using cancelable biometrics and puf in multi-server environment. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 5413–5428. DOI:10.1109/TIFS.2021.3128826
22. Hu Y, Bian W, Xie D, Xu D, Xu Z. Secure and efficient industrial wireless sensor networks protocol based on cancelable biometrics. *IEEE Trans. Ind. Inform.* **2024**, *20*, 13580–13590. DOI:10.1109/TII.2024.3423309
23. Dolev D, Yao A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. DOI:10.1109/TIT.1983.1056650
24. Zhang J, Qu G. Physical unclonable function-based key sharing via machine learning for iot security. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7025–7033. DOI:10.1109/TIE.2019.2938462

25. Ghammam L, Karabina K, Lacharme P, Thiry-Atighehchi K. A cryptanal ysis of two cancelable biometric schemes based on index-of-max hashing. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2869–2880. DOI:10.1109/TIFS.2020.2977533
26. Li Y, Pang L, Zhao H, Cao Z, Liu E, Tian J. Indexing-min–max hashing: Relaxing the security–performance tradeoff for cancelable fingerprint templates. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6314–6325. DOI:10.1109/TSMC.2022.3144854
27. Vigano L. Automated Security Protocol Analysis With the AVISPA Tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86. DOI:10.1016/j.entcs.2005.11.052
28. Jin Z, Lim MH, Teoh ABJ, Goi BM, Tay YH. Generating fixed-length representation from minutiae using kernel methods for fingerprint authentication. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 1415–1428. DOI:10.1109/TSMC.2015.2499725
29. Cappelli R, Maio D, Maltoni D, Wayman JL, Jain AK. Performance eval uation of fingerprint verification systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 3–18. DOI:10.1109/TPAMI.2006.20
30. Kong XJ, Li XJ, Jin Z, Zhou P, Chen JY. One-factor cancellable bio metrics verification scheme. *Acta Autom. Sin.* **2021**, *47*, 1159–1170. DOI: 10.16383/j.aas.c190059.
31. Li G, Yang B, Rathgeb C, Busch C. Towards generating pro tected fingerprint templates based on bloom filters. In Proceedings of the 3rd International Workshop on Biometrics and Forensics (IWBF 2015), Gjovik, Norway, 3–4 March 2015; pp. 1–6.
32. Gomez-Barrero M, Galbally J, Rathgeb C, Busch C. General framework to evaluate unlinkability in biometric template protection systems. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1406–1420. DOI:10.1109/TIFS.2017.2788000
33. Li CT, Hwang MS, Chu YP. A secure and efficient communication schemewith authenticated key establishment and privacy preserving for vehicular ad hoc networks. *Comput. Commun.* **2008**, *31*, 2803–2814. DOI:10.1016/j.comcom.2007.12.005